US006570853B1

(12) **United States Patent**
Johnson et al.

(10) **Patent No.: US 6,570,853 B1**
(45) **Date of Patent: May 27, 2003**

(54) **METHOD AND APPARATUS FOR TRANSMITTING DATA TO A NODE IN A DISTRIBUTED DATA PROCESSING SYSTEM**

(75) Inventors: **Stephen M. Johnson**, Colorado Springs, CO (US); **Timothy E. Hoglund**, Colorado Springs, CO (US); **David M. Weber**, Monument, CO (US); **John M. Adams**, Colorado Springs, CO (US); **Mark A. Reber**, Alpharetta, GA (US)

(73) Assignee: **LSI Logic Corporation**, Milpitas, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/679,506**

(22) Filed: **Oct. 4, 2000**

Related U.S. Application Data

(63) Continuation of application No. 09/216,401, filed on Dec. 18, 1998.

(51) **Int. Cl.**[7] **H04L 12/28**; G01R 31/08

(52) **U.S. Cl.** **370/236**; 370/412; 370/395.2; 370/400

(58) **Field of Search** 370/400, 428, 370/429, 235, 231, 230, 412, 413, 417, 236, 395, 410, 360, 395.2, 395.21, 236.1, 236.2, 411, 389, 392; 709/230–235

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,849,968 A		7/1989	Turner	370/94
4,928,096 A		5/1990	Leonardo et al.	
5,136,582 A		8/1992	Firoozmand	370/85.1
5,247,626 A		9/1993	Firoozmand	395/250
5,463,382 A		10/1995	Nikas et al.	
5,610,745 A		3/1997	Bennett	359/139
5,615,392 A		3/1997	Harrison et al.	395/876
5,644,575 A	*	7/1997	McDaniel	370/428
5,726,977 A	*	3/1998	Lee	370/235
5,740,466 A		4/1998	Geldman et al.	395/825
5,742,675 A		4/1998	Kilander et al.	
5,768,530 A		6/1998	Sandorfi	395/200
5,778,419 A		7/1998	Hansen et al.	711/112
5,781,801 A		7/1998	Flanagan et al.	395/876
5,784,358 A	*	7/1998	Smith et al.	370/230
5,856,972 A		1/1999	Riley et al.	
5,864,557 A		1/1999	Lyons	
5,914,936 A	*	6/1999	Hatono et al.	370/230
5,914,955 A		6/1999	Rostoker et al.	
6,000,020 A		12/1999	Chin et al.	
6,038,235 A		3/2000	Ho et al.	
6,091,710 A		7/2000	Mawhinney	
6,098,125 A		8/2000	Fiacco et al.	
6,101,166 A		8/2000	Baldwin et al.	
6,104,722 A		8/2000	Stewart	
6,128,283 A	*	10/2000	Sabaa et al.	370/236
6,185,203 B1	*	2/2001	Berman	370/360
6,188,668 B1		2/2001	Brewer et al.	
6,256,306 B1		7/2001	Bellenger	

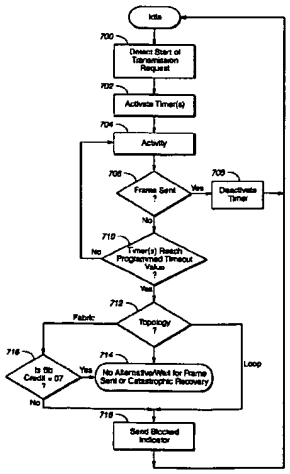
* cited by examiner

Primary Examiner—Hassan Kizou
Assistant Examiner—Hanh Nguyen
(74) *Attorney, Agent, or Firm*—Carstens, Yee & Cahoon LLP

(57) **ABSTRACT**

A method and apparatus in a source node for transmitting data to a target node. Responsive to a request to transmit data to the target node, a determination is made as to whether a selected period of time has passed without data transmitted from the source node being received by the target node. Responsive to detecting the selected period of time has passed without data transmitted from the source node being received by the target node, a determination is made as to whether space is available in the target node to receive the data. Responsive to a determination that space is unavailable in the target node, generating an indication that the target node is blocked is generated.

45 Claims, 6 Drawing Sheets



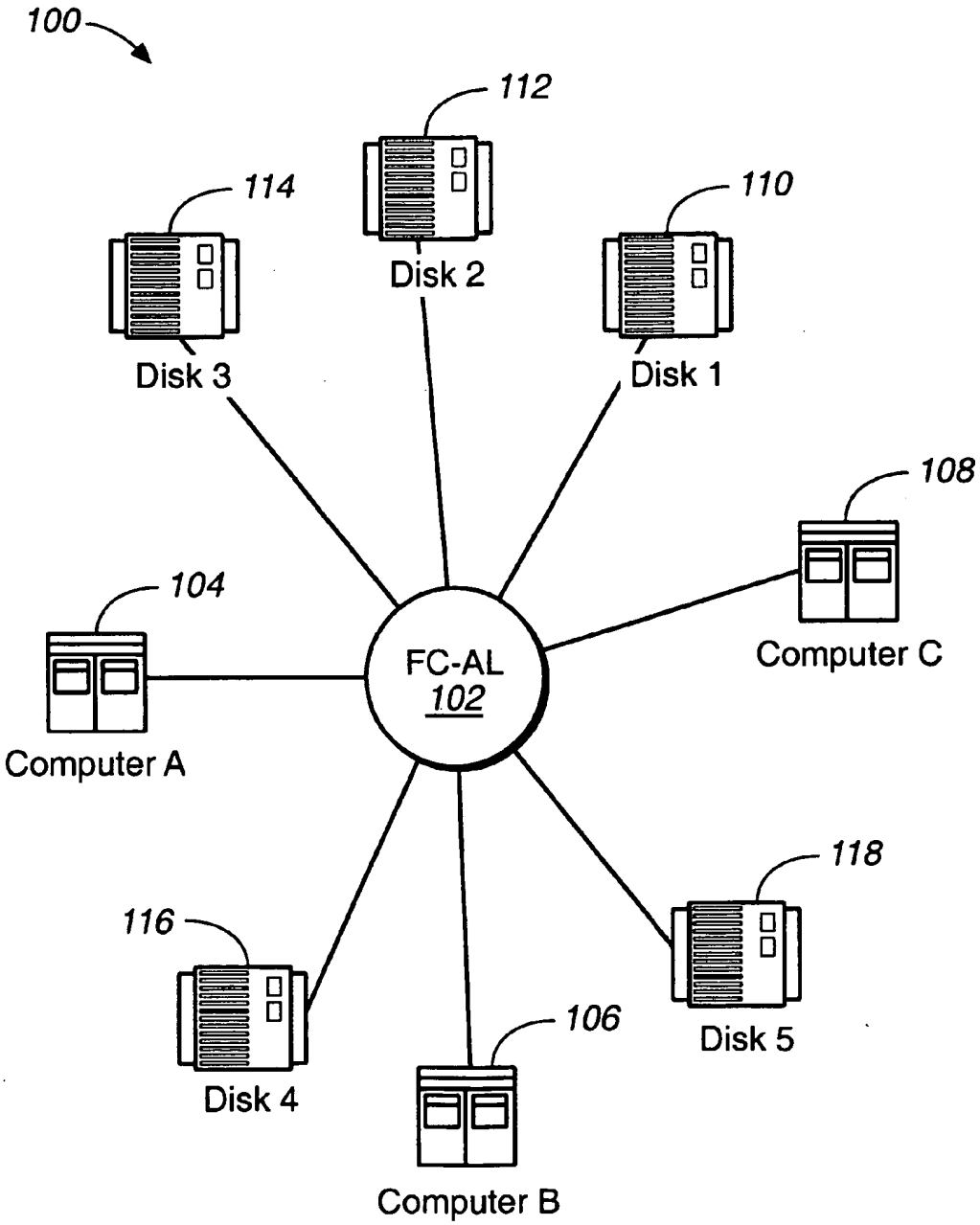

100
112
Disk 2
114
Disk 3
110
Disk 1
108
Computer C
104
Computer A
FC-AL
102
118
Disk 5
116
Disk 4
106
Computer B

FIG._1

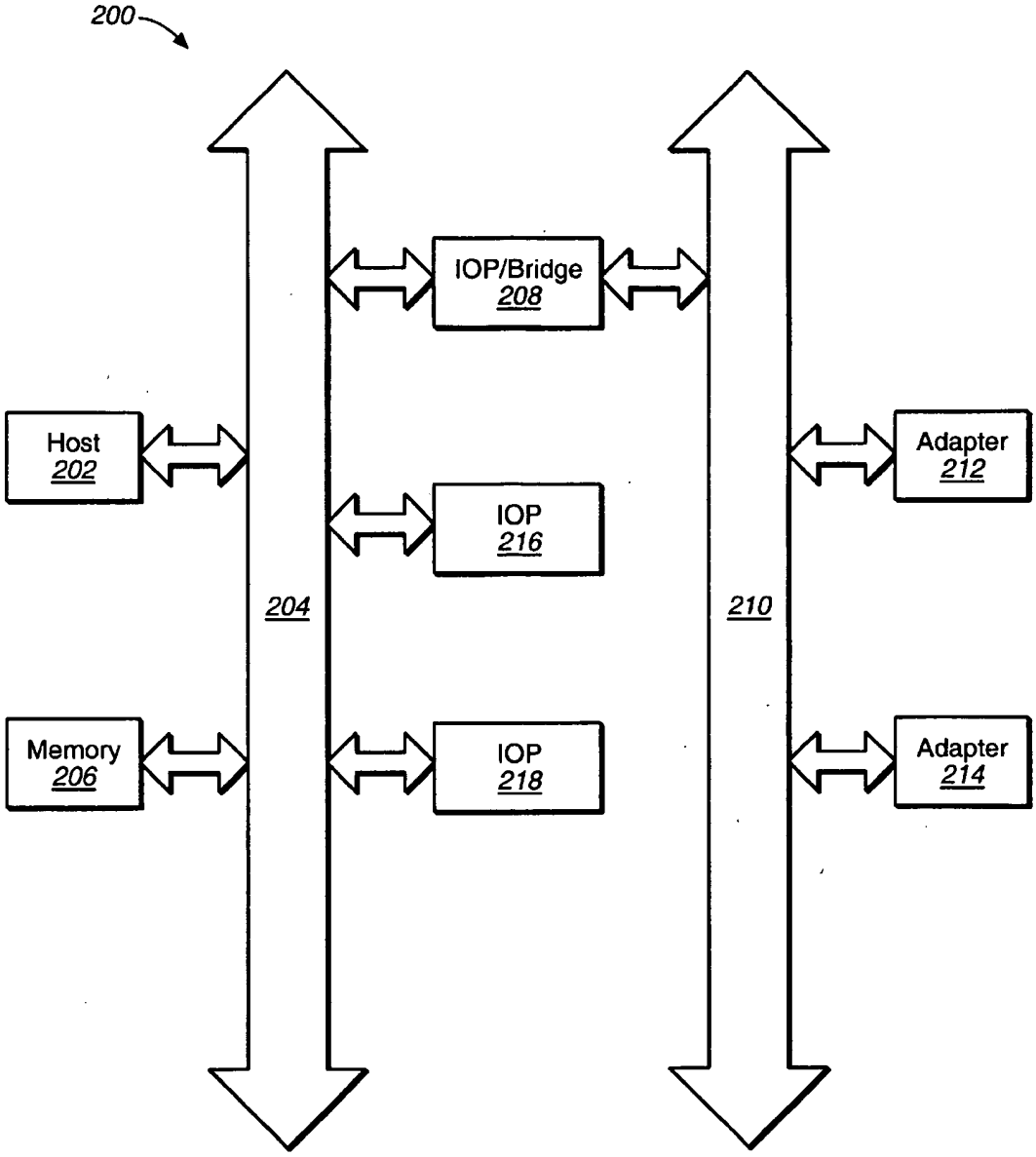

200
IOP/Bridge
208
Host
202
IOP
216
204
210
Adapter
212
Memory
206
IOP
218
Adapter
214

FIG._2

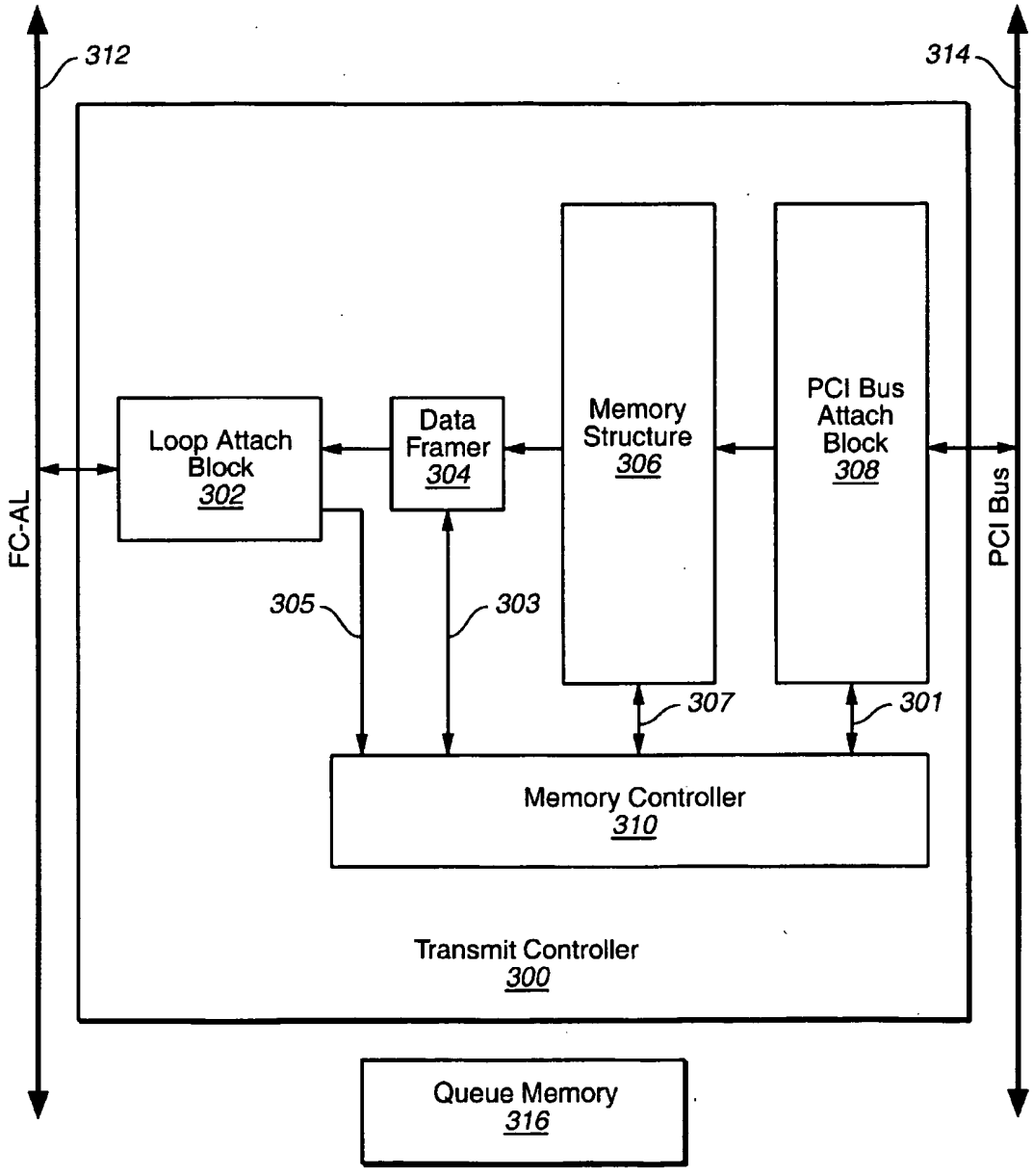

312
314
FC-AL
PCI Bus
Loop Attach Block 302
Data Framer 304
Memory Structure 306
PCI Bus Attach Block 308
305
303
307
301
Memory Controller 310
Transmit Controller 300
Queue Memory 316

FIG._3

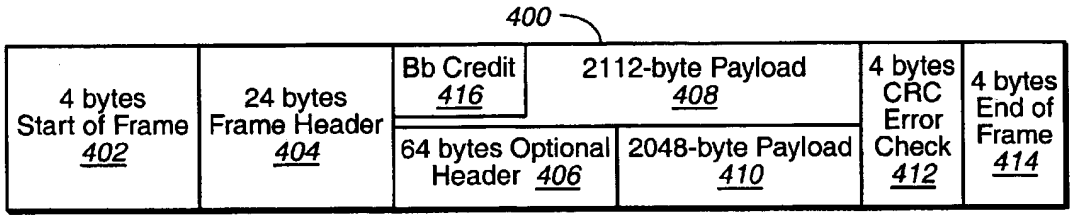

400
4 bytes Start of Frame 402
24 bytes Frame Header 404
Bb Credit 416
2112-byte Payload 408
64 bytes Optional Header 406
2048-byte Payload 410
4 bytes CRC Error Check 412
4 bytes End of Frame 414

FIG._4

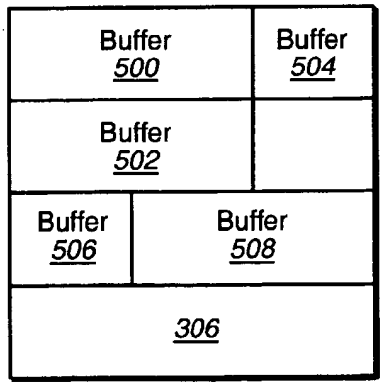

Buffer 500
Buffer 504
Buffer 502
Buffer 506
Buffer 508
306

FIG._5

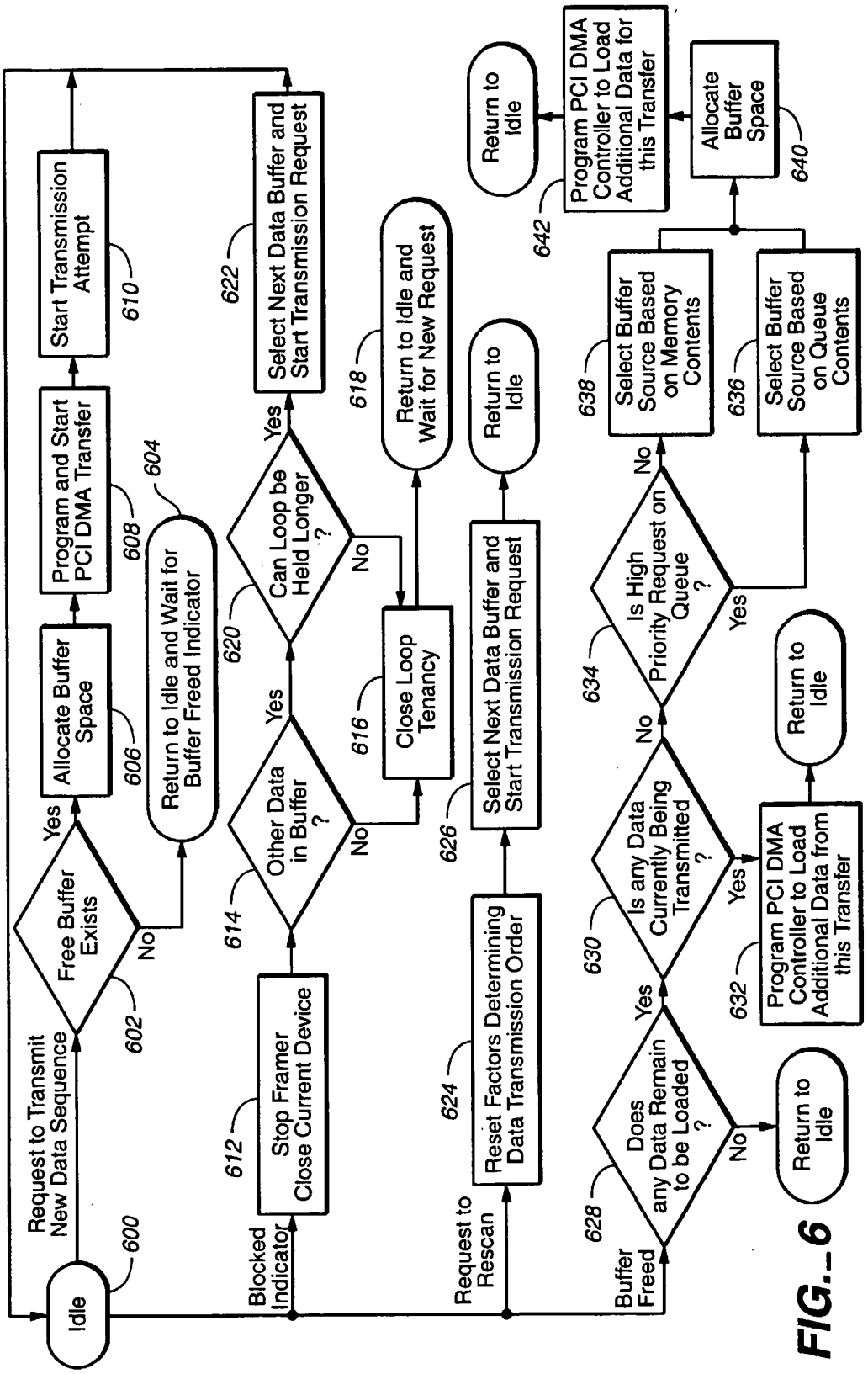

Idle
600
Request to Transmit New Data Sequence
Free Buffer Exists
602
Yes
No
Allocate Buffer Space
606
Program and Start PCI DMA Transfer
608
Start Transmission Attempt
610
Return to Idle and Wait for Buffer Freed Indicator
604
Blocked Indicator
Stop Framer Close Current Device
612
Other Data in Buffer ?
614
Yes
No
Close Loop Tenancy
616
Can Loop be Held Longer ?
620
Yes
No
Return to Idle and Wait for New Request
618
Select Next Data Buffer and Start Transmission Request
622
Request to Rescan
Reset Factors Determining Data Transmission Order
624
Select Next Data Buffer and Start Transmission Request
626
Return to Idle
Buffer Freed
Does any Data Remain to be Loaded ?
628
Yes
No
Return to Idle
Is any Data Currently Being Transmitted ?
630
Yes
No
Program PCI DMA Controller to Load Additional Data from this Transfer
632
Return to Idle
Is High Priority Request on Queue ?
634
Yes
No
Select Buffer Source Based on Queue Contents
636
Select Buffer Source Based on Memory Contents
638
Allocate Buffer Space
640
Program PCI DMA Controller to Load Additional Data for this Transfer
642
Return to Idle

FIG._6

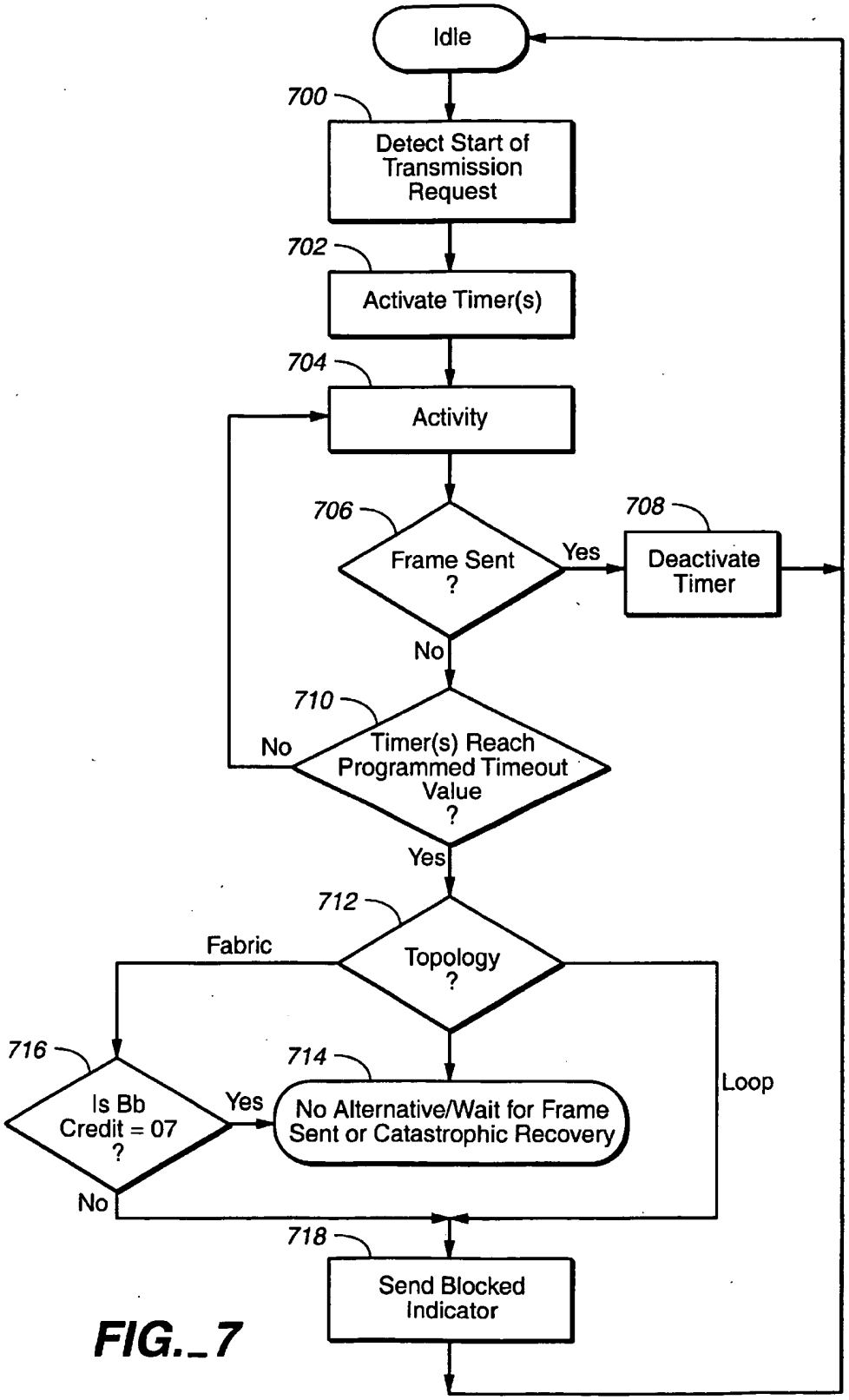

Idle
700
Detect Start of Transmission Request
702
Activate Timer(s)
704
Activity
706
Frame Sent ?
Yes
708
Deactivate Timer
No
710
Timer(s) Reach Programmed Timeout Value ?
No
Yes
712
Topology ?
Fabric
716
Is Bb Credit = 07 ?
Yes
714
No Alternative/Wait for Frame Sent or Catastrophic Recovery
Loop
No
718
Send Blocked Indicator

FIG._7

METHOD AND APPARATUS FOR TRANSMITTING DATA TO A NODE IN A DISTRIBUTED DATA PROCESSING SYSTEM

CONTINUATION IN PART

This application is a continuation in part of U.S. application Ser. No. 09/216,401, filed Dec. 18, 1998, titled Method and Apparatus for Transmitting Data.

BACKGROUND OF THE INVENTION

1. Technical Field

The present invention relates to an improved data processing system and in particular to a method and apparatus for transmitting data. Still more particularly, the present invention relates to a method and apparatus for managing transmission of data from a source to a destination.

2. Description of the Related Art

Two basic types of communications connections are employed between processors and between a processor and a peripheral. These types of connections are known as channels and networks. A channel provides a direct or switched point-to-point connection between communicating devices. This type of connection is typically employed between a processor and a peripheral device. The primary task of the channel is to transport data at the highest possible speed with the least delay. In contrast, a network is an aggregation of distributed nodes, such as workstations, file servers, and peripherals. Typically, in a network a node contends for the transmission medium and each node must be kept free of error conditions on the network. A traditional channel is hardware intensive and typically has lower overhead than a network. Conversely, networks tend to have relatively high overhead because they are software intensive. Networks, however, are expected to handle a more extensive range of tasks as compared to channels. In a closed system, every device addressed is known to the operating system either by assignment or pre-definition. This configuration knowledge is important to the performance levels of channels. Fibre Channel is a channel-network hybrid containing network features to provide the needed connectivity, distance, and protocol multiplexing along with enough traditional channel features to retain simplicity, repeatable performance, and guaranteed delivery. Fibre Channel has an architecture that represents a true channel/network integration. Fibre Channel allows for an active intelligent interconnections scheme, called a fabric, to connect devices. A Fibre Channel port manages simple point-to-point connection between itself and the fabric. A "port" is a hardware entity on a "node" with a node being a device connected to a network that is capable of communicating with other network devices. Transmission is isolated from control protocol. As a result, different topologies may be implemented. Fibre Channel supports both large and small data transfers.

The demand for flexible, high performance, fault-tolerant storage subsystems caused host adapter, disk storage, and high-capacity drive manufacturers to adopt Fibre Channel (FC) as a standard. This serial standard cuts cabling costs, increases data rates, and overcomes distance limitations commonly associated with a Small Computer System Interface (SCSI). Fibre Channel can carry SCSI protocols, and as a result offers an ideal upgrade for work stations, servers, and other systems requiring high availability and/or high bandwidth. Fibre Channel has become increasingly important as companies are seeking to provide faster and easier access to data for various clients. The Fibre Channel Standard (FCS) as adopted by the American National Standards Institute (ANSI), provides a low cost, high speed interconnect standard for workstations, mass storage devices, printers, and displays.

Current Fibre Channel data transfer rates exceed 100 megabytes (Mbytes) per second in each direction. Fibre Channel data transfer rates also may be scaled to lower speed, such as 50 Mbytes per second and 25 Mbytes per second. This technology provides an interface that supports both channel and network connections for both switched and shared mediums. Fibre Channel simplifies device interconnections and reduces hardware cost because each device requires only a single Fibre Channel port for both channel and network interfaces. Network, port to port, and peripheral interfaces can be accessed though the same hardware connection with the transfer of data of any format.

In sending data from a source node to a destination node, the source transmits data from a bus, such as a Peripheral Component Interconnect (PCI) bus, to a buffer for transfer onto a Fibre Channel system, which is connected to the destination node. Data is sent serially on Fibre Channel systems. As a result, data currently in a buffer must be sent before additional data may be loaded. Currently, if data cannot be sent because the destination is not accepting additional data, then this data must be removed to send data to another destination. This loading and dumping of data increases the overhead in transferring data between various nodes on a Fibre Channel system.

Additionally, in some cases, a source node will attempt to transmit data to a destination node on a network that has receive buffers that are full or unable to receive data. In such a situation, presently known devices will continue to attempt to send data to this node. In the case of a Fibre Channel system that includes an arbitrated loop, such a device will continue to hold the arbitrated loop and freeze the entire network, preventing data transfers by other devices to other destinations.

Thus, it would be advantageous to have an improved method and apparatus for transferring data between nodes in which the reduction in network efficiency, caused by a device continuing to transmit data to a node that is not accepting data, is reduced.

SUMMARY OF THE INVENTION

The present invention provides a method and apparatus in a source node for transmitting data to a target node. Responsive to a request to transmit data to the target node, a determination is made as to whether a selected period of time has passed without data transmitted from the source node being received by the target node. Responsive to detecting the selected period of time has passed without data transmitted from the source node being received by the target node, a determination is made as to whether space is available in the target node to receive the data. Responsive to a determination that space is unavailable in the target node, an indication that the target node is blocked is generated.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

FIG. 1 is a diagram of a data processing system in which a preferred embodiment of the present invention may be implemented;

FIG. **2** is a block diagram of a data processing system in accordance with a preferred embodiment of the present invention;

FIG. **3** is a block diagram of a transmit controller used to transfer data in accordance with a preferred embodiment of the present invention;

FIG. **4** is a diagram of a frame handled by the present invention;

FIG. **5** is a diagram illustrating allocation of buffers in a memory structure in accordance with a preferred embodiment of the present invention;

FIG. **6** is a flowchart of a process for managing a buffer in a node in accordance with a preferred embodiment of the present invention; and

FIG. **7** is a flowchart of a process for detecting blocked transmission of data in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION

With reference now to FIG. **1**, a diagram of a data processing system is illustrated in which a preferred embodiment of the present invention may be implemented. Data processing system **100** incorporates a network on the form of a Fibre Channel fabric **102**. In this example, Fibre Channel **102** is a Fibre Channel arbitrated loop (FC-AL). Although the depicted example involves a fabric in the form of an arbitrated loop, the present invention may be applied to other fabrics, such as, for example, a point to-point or switched fabric. In a point-to-point fabric, if blocking occurs, nothing else can be done. With a switched fabric, the process and decision making are the same, but the events that will cause the blocking indication are different.

Still with reference to FIG. 1, computer **104**, computer **106**, and computer **108** are connected to fabric **102**. In addition, disk storage unit **110**, disk storage unit **112**, disk storage unit **114**, disk storage unit **116**, and disk storage unit **118** also are connected fabric **102**. The various computers, computers **104–108**, may access data located on the various disk storage units, disk storage units **110–118**. Of course, other devices in computers may be connected to fabric **102** depending on the implementation. In this topology, a node, such as computer **108** may send data to a target node such as computer **104** or disk storage unit **116**. Typically, computer **108**, as the source node, will place data in a buffer for transmission to a target node. If the target node is not excepting data, an indication will be received at computer **108** that the transfer has been blocked by the target node. In this instance, the shared fabric resources (source node) may be held idle waiting for the target to indicate that it will accept data or the transfer will be aborted with the data being dumped or cleared to make way for another transfer to another node. At a later time, computer **108** may again try sending data to the target node by reloading the data and attempting to retransmit the data to the target node. A Fibre Channel arbitrated loop topology as shown for Fibre Channel fabric **102** allows for multiple communicating ports to be attached in a loop without requiring hubs or switches. The loop is a shared-bandwidth distributed topology where each port includes the minimum necessary connection function. A port may arbitrate or use an arbitrated loop. Once a port wins the arbitration, based on the lowest port address, a second port may be opened to complete a single bi-directional point-to-point circuit. With the loop, only one pair of ports may communicate at one time. When two connected ports release control of the loop, another point-to-point circuit may be established between two ports. FIG. **1** is intended as an example of a distributed data processing system in which the processes and apparatus of the present invention may be implemented, and not as an architectural limitation for the present invention.

The present invention provides a mechanism that avoids this situation by retaining the data within the buffer while new data is loaded into another buffer for transfer to another target node. Further, multiple sets of data may be loaded into buffers using the mechanism of the present invention. Fibre Channel fabric **102** is scanned for a node that will accept data. The scanning may be performed in a number of different ways, such as, for example, attempting transmission to nodes in an ordered list or using a round robin scheme, which is a sequential, cyclical selection of target nodes.

In addition, the present invention provides a mechanism to indicate a situation in which a destination is unlikely to accept data from the target. This mechanism includes using a timer to identify when too much time has passed for transmission of data to a target. In addition, the mechanism will produce a signal that is used to indicate that the transfer should be halted. In such an instance, another function or transfer may then be started. The function or transfer that is started may continue to use Fibre Channel fabric **102** or may release the fabric for another node to use when Fibre Channel fabric **102** is in the form of an arbitrated loop.

Turning next to FIG. **2**, a block diagram of a data processing system is depicted in accordance with a preferred embodiment of the present invention. Data processing system **200** includes a host **202**, which may contain one or more processors, which form the central processing unit (CPU) or CPUs for data processing system **200**. Data processing system **200** is a data processing system designed along the Intelligent Input/Output (I_2O) Architecture Specification, version 1.5, March 1997 available from the I_2O Special Interest Group. The present invention, however, may be implemented using other system architectures.

The processors within host **202** may be, for example, a Pentium II processor operating at 400 Mhz, which is available from Intel Corporation in Santa Clara, Calif. In the depicted example, primary bus **204** and secondary bus **210** are PCI buses although the present invention may be implemented using other types of buses.

Still referring to FIG. **2**, data processing system **200** includes a primary input/output platform (IOP) **208**, which is connected to host **202** by primary bus **204**. In data processing system **200**, memory **206** is attached to primary bus **204**. Additionally, IOP **208** is connected to secondary bus **210** and also functions as a PCI-to-PCI bus bridge. Data processing system **200** also includes adapter **212** and adapter **214**. Secondary IOPs **216** and **218** are intelligent adapters under I_2O and contain input/output processors. Adapters **212** and **214** are non-intelligent adapters, which do not contain input/output processors. The processes and apparatus of the present invention may be implemented in the various adapters and IOPs in data processing system **200**.

Turning now to FIG. **3**, a block diagram of a transmit controller used to transfer data is depicted in accordance with a preferred embodiment of the present invention. Transmit controller **300** is an example of a transmit controller that may be found within an IOP or an adapter in a data processing system, such as data processing system **200** in FIG. **2**. Transmit controller **300** may be implemented in a number of ways, such as, for example, as an application specific integrated circuit (ASIC). Transmit controller **300** includes a loop attach block **302**, a data framer **304**, a

memory structure **306**, a PCI bus attach block **308**, and a memory controller **310**. Loop attach block **302** is an interface that provides a connection to a Fibre Channel fabric **312** to transmit and receive data from Fibre Channel fabric **312**. In the depicted example, Fibre Channel fabric **312** is a Fibre Channel arbitrated loop. Loop attach block **302** provides 8b/10b encode/decode functions and reorders bytes in the appropriate manner to be compatible with an external serializer/deserializer. The external serializer/deserializer (SERDES) converts the data between the serial differential pair bit stream used by Fibre Channel **312** and the parallel data used by loop attach block **302**. Data framer **304** is responsible for routing data from memory structure **306** and adding any required information to generate a frame for transfer onto Fibre Channel **312**. A frame is the smallest indivisible packet of data that is sent onto a Fibre Channel system. Depending on the type of network used, data framer **304** may create other types of frames or data packets.

Turning to FIG. **4**, a diagram of a frame handled by the present invention is illustrated. Frame **400** includes a start of frame delimiter **402**, a frame header **404**, an optional header **406**, along with payloads **408** and **410**. Frame **400** also includes a 32 bit CRC **412** for error detection and an end of framer delimiter **414**. In the case of a logon frame, an identification of buffer to buffer (Bb) credit **416** is placed in payload **408** in frame **400**. In the depicted example, each frame or group of frames is acknowledged when received by a target node. This acknowledgement also provides notification of non-delivery of a frame to a target node. Reception of a frame may be acknowledged depending on what is called the "class of service". Class **1** service and **2** service provide for an acknowledgement called an "ACK". Class **3** service (the most popular as of today) does not provide for such an acknowledgement. For this class, it is up to the protocol using the Fibre Channel connection to provide a method to determine if data was successfully delivered. A node does provide a signal called an R_RDY to indicate when it has cleared a buffer and is able to receive another frame. The only indication that no buffers are available is the lack of this signal.

Turning back to FIG. **3**, memory structure **306** contains the data that is to be transferred to various destination nodes on the Fibre Channel arbitrated loop. PCI bus attach block **308** is employed to pass information across PCI bus **314**. Memory controller **310** contains the processes and logic of the present invention used to intelligently transfer data and manage memory structure **306**.

Memory controller **310** begins the transfer of data to a target node by sending the appropriate signals to PCI bus attach block **308** through PCI load control line **301** to load data from a host memory on PCI bus **314**. The data is loaded into memory structure **306**. Thereafter, memory controller **310** sends a start instruction to data framer **304** through start/abort control line **303** to route data to the appropriate target node. Blocked indicator line **305** is used to provide a blocked indicator, which is used by memory controller **310** to potentially stop data framer **304** and release the fabric.

Memory structure **306** may be divided into a number of different buffers in which data to be transferred to a target node is to be stored. In the depicted example, memory structure **306** is configured to be large enough to store pre-loaded transmission data in a manner that deletion of this data is avoided when the transmission of data is not possible due to a lack of received buffers or full-received buffers on a target node. Of course, the various buffers within memory structure **306** need not be contiguous. When the transfer of a set of data is not possible because the destination, a target node, is not receiving data, additional data may be loaded into a different location in memory structure **306** destined for a different node. In the depicted example, the additional data may be loaded into another buffer allocated within memory structure **306**. A queue memory **316** is present for storing transfer requests. Queue memory **316** could be located on chip, off chip but on the adapter board, or in host memory. Queue memory **316** is simply a shared storage element (in this case a memory region) accessible by the processor generating the transfer requests and memory controller **310**. A common data structure is defined to allow transfer requests to be communicated between the two processing elements.

Turning next to FIG. **5**, a diagram illustrating allocation of buffers in a memory structure is depicted in accordance with a preferred embodiment of the present invention. As can be seen, memory structure **306** includes buffers **500–508**. Each of these buffers is allocated for a set of data, such as frames for transmission to a target node. In this example, five sets of data are stored within memory structure **306**. Although the buffers shown in memory structure **306** are shown as contiguous blocks of memory, contiguous allocation of memory for a buffer is not required. Each of the sets of data are destined for a particular node on a Fibre Channel arbitrated loop (FC-AL). Memory controller **310** directs the order in which these sets of data are transmitted to the target nodes. Data may be loaded for a new transfer each time a target node is not accepting data. Alternatively, multiple sets of data may be loaded into memory structure **306** for transfer based on various mechanisms, such as a priority scheme or a round robin scheme. All currently loaded data may be scanned until a node is found that has the capacity to receive the data or is held idle as above in the Fibre Channel fabric. Previously, when a destination node was not receiving data, the information in the buffer was unloaded and another set of data destined for another node was loaded for transmission. In the depicted example, the set of data for a particular node is retained within memory structure **306**, while memory controller **310** loads a new set of data destined for another node into a buffer in memory structure **306**. When multiple sets of data are located within memory structure **306**, memory controller **310** may scan the Fibre Channel fabric to decide which node may receive data. In scanning the Fibre Channel fabric for a node to accept data, different strategies may be employed depending on the implementation. One strategy involves proceeding through a list of nodes and attempting to send data to each node on the list in the order encountered. Another strategy involves using a round robin process for selecting nodes for data transfer.

When transfer of data begins, memory controller **310** sends a start signal to data framer **304** through start/abort control line **303** to begin transferring data from memory structure **306** to loop attach block **302** for ultimate transfer onto the Fibre Channel fabric. Memory controller **310** will halt the transfer of data to a particular node on the Fibre Channel in response to receiving a blocked indicator from loop attach block **302** through blocked indicator line **305**. In response to such a signal, memory controller **310** will send an abort control signal to data framer **304** to stop transfer of data. Memory controller **310** also will send load signals to PCI bus attach block **308** to initiate the loading of additional sets of data in response to the set of data being sent to the destination node or in response to an inability to transfer data to the destination node. Memory controller **310** selects the various sets of data for transfer in memory structure **306** through various buffer selection signals sent on buffer selection line **307**.

With reference now to FIG. **6**, a flowchart of a process for managing a buffer in a node is depicted in accordance with a preferred embodiment of the present invention. The process in FIG. **6** illustrates the process in states and steps employed by memory controller **310** in transferring data to target nodes on a Fibre Channel arbitrated loop. The process begins in an idle state **600** waiting for a signal. In response to receiving a request to transmit a new data sequence, a determination is made as to whether free buffer space exists in the memory structure (step **602**). If free buffer space does not exist in the memory structure, the process return to the idle state **600** and waits for a buffer freed indicator (step **604**). Otherwise, buffer space is allocated in the memory structure (step **606**), and program and start PCI DMA transfer occurs (step **608**). A program and start direct memory access (DMA) request consists of obtaining a scatter/gather list element from the transfer request data structure in the queue memory and providing it to the PCI bus attach block **308**. (A scatter/gather list consists of a set of address/byte-count pairs that completely describe the memory location(s) from which data for the transfer is to be obtained. A start signal is then sent to the PCI bus attach block **308** indicating that the DMA programming is complete and the PCI bus attach block **308** may execute the data transfer. Thereafter, a start transmission attempt is made (step **610**) with the process then returning to idle state **600**. A start transmission request is a simple handshake indicating to data framer **308** that the correct data is located in the memory structure and that all other information required for the data framer **308** to build the proper frames has also been programmed into data framer **308** by memory controller **310**. The result of this step is a request sent to loop attach block **302** to send the frame.

In response to receiving a blocked indicator, data framer **304** is stopped and the current device is closed (step **612**). This step is used to halt transfer of data when an indication is received that the target node is blocked or unable to receive data from the source node. This mechanism in the case of an arbitrated loop will prevent a device from maintaining the loop in an idle state.

In the depicted examples, the term "blocked" refers to a condition in which buffer to buffer credit has not been granted for a particular period of time. Other indicators are present when a frame can not be sent, such as P_BSY, which indicates that the port is busy. These types of indicators are well defined in specifications for FC-AL and Fibre Channel-Physical and Signaling Interface (FC-PH). The responses for these other types of indications are defined within the specifications. Buffer to buffer credit is used as a mechanism for each node on a loop or network to determine the number of buffers on its associated attached port. This mechanism includes a log-on process that occurs when the network is initialized. The ports exchange frames in which the payloads contain the buffer to buffer credit information. Additionally, buffer to buffer credit is how the interchange of frames and R_RDYs are performed. Basically, the R_RDY indicates a freed buffer. Each port sends a R_RDY when a buffer is freed. In addition, each port contains a counter as to the number of freed buffers on the other port that is initialized to the log-on value, decremented for each frame sent, and incremented for each R_RDY received. This counter must be non-zero to send a frame.

Explanation as to how the blocked indicator is obtained is described with respect to the description of FIG. **7** below. In this step, the memory controller sends a stop signal to the framer to stop sending data on to the Fibre Channel fabric. In addition, the current device is closed. Closing a "device", also called "CLS", refers to a signal being sent to the target node to indicate that no further data is to be sent. The target node will then acknowledge this by repeating the CLS itself. At this time, the source node may open another target node.

Then, a determination is made as to whether other data is stored in the buffer (step **614**). If other data is not stored in the buffer, then close loop tenancy is performed (step **616**). Tenancy refers to the period of time from a node achieving successful arbitration to the time the node no longer needs the loop (i.e. other nodes may win arbitration). Closing the loop refers to the situation in which a node no longer desires to open another target node and will allow another node to win arbitration and use the loop if so desired. In this step, the loop is released. Thereafter, the process returns to idle and waits for a new request (step **618**). A transmit new data request may also be received. (i.e. the controller may continue to load new data if buffers are available even though it is not currently transmitting data).

With reference again to step **614**, if other data is stored in the buffers in memory structure **306**, a determination is made as to whether the loop can be continued to be held by the node (step **620**). The decision in step **620** may be based on the likelihood that more data can be sent, the time the loop has already been held, and other factors depending on the implementation. For example, after a single attempt to send each buffer once, attempts may be stopped. Alternatively, attempts to transfer data may be made in a round robin order for a set period of time before the loop is released. All high priority traffic may be continuously attempted until sent, then a low priority attempt is sent. If the loop cannot be continued to be held, the process then proceeds to step **616** to release the loop. Otherwise, the next data in the buffer in memory structure **306** is selected for transfer and a start transmission request is made (step **622**). The selection of the next data buffer in memory structure **306** may be based on a number of different mechanisms. For example, the next data buffer may be selected based on the number of attempts already made to send data to the buffer, the priority of buffer traffic and other factors. The selection may be made in the order of queuing or on some priority order associated with each set of data. In addition, the selection also may be based on the total number of buffers in memory structure **306**. The process then returns to the idle state **600**.

In response to receiving a request to rescan, factors determining data transmission order are reset (step **624**). Then, the next data buffer is selected and a start transmission request is made (step **626**) with the process returning to the idle state **600**.

In response to receiving a buffer freed signal, a determination is made as to whether data remains to be loaded into memory structure **306** (step **628**). If data does not remain to be loaded, the process then returns to idle state **600**. Otherwise, a determination is made as to whether data is currently being transmitted (step **630**). If data is currently being transmitted, then the PCI DMA controller is programmed to load additional data for this transfer (step **632**) with the process then returning to the idle state **600**.

With reference again step **630**, if data is not currently being transmitted, a determination is then made as to whether a high priority request is located on the queue (step **634**). If a high priority request is located on a queue then, a buffer source is selected based on the queue contents (step **636**). If a high priority request is not located on the queue, then a buffer source is selected based on the memory contents (step **638**). In both instances, buffer space in memory structure **306** is allocated after a buffer source is

selected (step **640**). Thereafter, the PCI Bus Attach block is programmed to load the additional data for the transfer (step **642**) with the process then returning to idle state **600**.

Turning next to FIG. **7**, a flowchart of a process for detecting blocked transmission of data is depicted in accordance with a preferred embodiment of the present invention. The process in FIG. **7** illustrates the process employed by loop attach block **302** in transferring data to target nodes on a Fibre Channel arbitrated loop. This process interacts with the process in FIG. **6** to provide a blocked indicator for use by memory controller **310** when appropriate.

The process begins by detecting the start of a transmission request (step **700**). This step may typically involve a request that the loop attached send a frame and the operation may be as simple as sending a frame immediately after another. The operation may be as complex as arbitrating for the loop and then waiting for credit to send a frame. The timer may be started at any point in the operation depending on the implementation. For example, the implementation may decide not to count the time to request to arbitrate when determining if a frame has been blocked. Additionally, the timer may be activated by a other transmission activities. For example, mere acquisition of the loop is an example of another transmission activity that may be used to activate the timer in step **700**. Upon detecting the start of a transmission request, the timers are then activated (step **702**).

Additionally, the timer may be activated when a node has arbitrated for an acquired loop in the instance that the topology is an arbitrated loop topology. This mechanism for activating a timer times the efforts to receive appropriate credit from the destination node after the initial arbitration is completed. Alternatively, the timer may be activated when the transmitter in the source node requests a frame to be transmitted. In such an instance, the timer is run only when the loop is acquired (i.e., the node has one arbitration for the loop). This process ignores the time taken to arbitrate access for the loop and only counts the time that the node is using the loop bandwidth. In such an instance, arbitration may be required multiple times until a credit is returned from the destination node. Thereafter, the process waits for an activity to occur (step **704**). Upon the occurrence of an activity, a determination is made as to whether the activity is the sending of a frame (step **706**). If the activity is the sending of a frame, the process then deactivates the timer (step **708**) with the process then returning to step **700**. Otherwise, a determination is made as to whether the timers have reached a programmed timeout value (step **710**). If the timers have not reached a program timeout value, the process returns to step **704** to wait for the occurrence of an activity. Timer values may vary depending on the implementation. One example in which a timer value may be set is the use of a timer in the form of a counter that counts all times starting from when a transmission request is asserted. The value may have a 1-microsecond resolution and would be programmable from 1 to 255 microseconds with an expected value of 10 microseconds. Multiple timers may be required when some values are continuously running when others are starting and stopping for different connections. As a result, different counter values would be required that are only incremented when the appropriate enable conditions are true for the different connections.

If the timer has reached a programmed timeout value, then an identification of the topology of the system is made (step **712**). If the topology is a point to point topology, the process then proceeds to wait for a frame to be sent or a catastrophic recovery (step **714**). This condition does not result in the blocked condition because nothing can occur at this point. It is possible, however, that data corruption on the media may result in R_RDYs that have not been properly received, resulting in the two ports not being in the consistent state, i.e., one is waiting for credit but the other has sent it with the credit being un-received. In such a situation, the timeout in recovery from this condition is defined in FC_AL and FC_PH. If the topology is a fabric topology, a determination is then made as to whether the buffer to buffer (Bb) credit is equal to zero (step **716**). If the buffer to buffer credit is equal to zero, the process also proceeds to step **714**. Otherwise, the process sends a blocked indicator (step **718**). This blocked indicator is sent to the process illustrated in FIG. **6** to generate a blocked indicator signal from idle state **600** in FIG. **6**.

With reference again to step **712**, if the topology is a loop topology, then the process proceeds directly to step **716** as described above.

A number of rules may be used in managing the timers in FIG. **7**. For example, a timer may be reset when a frame is queued and indicated a block condition when a program value is reached. Additionally, a timer may be reset when a node has first accessed the network and has indicated a blocked condition when a program value is reached. Also, a timer may be reset, but run only when the node is actually accessing the network. This situation is more complex, but only counts the time that a node is using network resources.

In a point to point network, a network in which two nodes are directly connected, the transfer of data is governed by a buffer to buffer credit that controls the transfer of a frame from one buffer to the next as opposed to an end to end credit, which controls the transfer of frames from a source to a destination through a network. If a frame can not be transferred to the next buffer, then no decision is possible because no other frame could be transferred either. Buffer to buffer credit is used to indicate whether the next buffer memory has room for the frame that is to be sent towards a destination. End to end credit indicates whether the destination node has enough room to accept the particular frame. The identification of end to end credit is useful in a fabric topology. In the decision in step **716**, if the buffer to buffer credit is equal to zero, then the next buffer is full, but the frame could be sent at a later time. If the buffer to buffer credit is equal to one, then the reason that the frame has not been sent is probably an end to end credit problem.

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms, and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media such a floppy disc, a hard disk drive, a RAM, and CD-ROMs and transmission-type media such as digital and analog communications links.

The description of the present invention has been presented for purposes of illustration and description, but is not limited to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. For example, although the processes used to generate a blocked indicator in the depicted example are shown in a loop attach block, they may be implemented in other places. For example, a fixed hardware block with no programmability may be used to implement the processes of the present invention. In the

depicted example, memory controller **310** is an embedded reduced instructions set computing (RISC) core that also may implement the processes of the present invention. The timer counters may be implemented in a programmable micro-sequencer and be programmed to detect the blocked condition using an external time base. Alternatively, an on-board processor in the adapter or elsewhere may be used to perform the entire function. Alternatively, some functions may be placed in hardware while other functions in software.

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media such a floppy disc, a hard disk drive, a RAM, and CD-ROMs and transmission-type media such as digital and analog communications links.

The description of the present invention has been presented for purposes of illustration and description, but is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated. While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.

What is claimed is:

1. A method for transmitting data from a source to a target node, wherein the target node has a buffer, the method comprising:

detecting a request to transmit data from the source node to the target node;

responsive to detecting the request to transmit data to the target node, activating a timer;

responsive to detecting an expiration of the timer, determining whether the buffer is available to receive the data, and

responsive to determining whether the buffer is available, generating a buffer status signal.

2. The method of claim **1**, wherein a buffer to buffer credit is used to transmit data and wherein the step of determining whether the buffer is available comprises:

responsive to detecting an expiration of the timer, determining whether buffer to buffer credit is present in the target node.

3. The method of claim **2**, wherein the status signal is a blocked signal when a determination is made that the buffer is unavailable.

4. The method of claim 1 further comprising:

responsive to the buffer status signal being a blocked status signal, halting attempts to transmit data to the target node.

5. The method of claim **4**, wherein the data is a first set of data and further comprising:

responsive to the blocked status signal, loading a second set of data into the memory for transmission to another target node, while retaining the first set of data in the memory.

6. The method of claim **1**, wherein the request to transmit data comprises a request to send a frame of data to the target node.

7. The method of claim **1**, wherein the source node is connected to the target node by a loop and wherein the request to transmit data comprises arbitrating for the loop.

8. The method of claim **1**, wherein the source node is connected to the target node by a network.

9. The method of claim **1**, wherein the source node is connected to the target node by a Fibre Channel fabric.

10. The method of claim **9**, wherein the Fibre Channel fabric is an arbitrated loop.

11. The method of claim **9**, wherein the Fibre Channel fabric is a switched fabric.

12. The method of claim **1**, wherein the source node is a computer.

13. The method of claim **1**, wherein the source node is an adapter.

14. A method for transmitting data from a source node to a target node, the method comprising:

responsive to detecting a request to transmit data to the target node, activating a timer;

responsive to detecting an expiration of the timer, determining whether space is available in the target node to receive the data; and

responsive to a determination that space is unavailable in the target node, generating an indication that the target node is blocked.

15. The method of claim **14**, wherein the request to transmit data comprises a request to send a frame of data to the target node.

16. The method of claim **14**, wherein the source node is connected to the target node by a loop and wherein the request to transmit data comprises arbitrating for the loop.

17. The method of claim **14**, further comprising:

responsive to the indication, halting attempts to transmit data to the target node.

18. The method of claim **14**, wherein a determination of determining whether space is available in the target node to receive the data is made by examining buffer to buffer credit from the target node.

19. A method in a source node for transmitting data to a target node, the method comprising:

detecting a request to transmit data from the source node to the target node;

responsive to detecting the request to transmit data to the target node, activating a timer;

monitoring the timer for an expiration of the timer;

responsive to detecting an expiration of the timer, determining whether a buffer to buffer credit is present in the target node; and

responsive to a determination that buffer to buffer credit is absent, generating a blocked signal.

20. An apparatus comprising:

a memory, wherein the memory is configured to hold data for transmission to target node;

a controller, wherein the controller controls transmission of data located in the memory to the target node;

an interface connected to the memory and configured to connection to a data channel,

wherein the interface has a plurality of modes of operation including:

a first mode of operation in which the interface monitors for a request to transmit data signal;

a second mode of operation, responsive to detecting a request to transmit data signal, in which the control-

ler activates a timer and determines whether the timer expires;

a third mode of operation, responsive to detecting an expiration of the timer, determining whether space is available in the target node to receive the data; and

fourth mode of operation, responsive to a determination that space is unavailable in the target node, in which the controller generates an indication that the target node is blocked.

21. The apparatus of claim **20**, wherein the memory is connected to the interface by a data framer, which creates frames from the data for the data transfer.

22. The method of claim **20**, wherein the controller has a plurality of modes of operation including:

a first mode of operation in which the controller is idle and monitors for signals;

a second mode of operation, responsive to detecting a request to transmit data signal in the first mode of operation, in which the controller determines whether space is present in the memory and allocates a first buffer space in the memory for a first data transfer, loads a set of data corresponding the request into a buffer space in the memory, and begins the data transfer; and

a third mode of operation, responsive to generation of the blocked indication by the interface while a data transfer is occurring, in which the controller halts the data transfer, determines if another set of data is available for transfer and begins another data transfer using the another set of data.

23. The apparatus of claim **20**, wherein other data is present in another buffer and wherein the controller further includes:

a fourth mode of operation, responsive to detecting a request to rescan in the first mode of operation, in which the controller selects the another buffer for transmission and begins transmission of the other data.

24. The apparatus of claim **20**, wherein the data channel is a Fibre Channel arbitrated loop.

25. The apparatus of claim **20**, wherein the data channel is a switched fabric.

26. A source node for transmitting data to a target node, wherein the target node has a buffer, the source node comprising:

detecting means for detecting a request to transmit data from the source node to the target node;

activating means, responsive to detecting the request to transmit data to the target node, for activating a timer;

determining means, responsive to detecting an expiration of the timer, for determining whether the buffer is available to receive the data; and

generating means, responsive to determining whether the buffer is available, for generating a buffer status signal.

27. The source node of claim **26**, wherein a buffer to buffer credit is used to transmit data and wherein the determining means whether the buffer is available comprises:

determining means, responsive to detecting an expiration of the timer, for determining whether buffer to buffer credit is present in the target node.

28. The source node of claim **27**, wherein the status signal is a blocked signal when a determination is made that the buffer is unavailable.

29. The source node of claim **26**, further comprising:

halting means, responsive to the buffer status signal being a blocked status signal, for halting attempts to transmit data to the target node.

30. The source node of claim **29**, wherein the data is a first set of data and further comprising:

loading means, responsive to the blocked status signal, for loading a second set of data into the memory for transmission to another target node, while retaining the first set of data in the memory.

31. The source node of claim **29**, wherein the source node is a computer.

32. The source node of claim **29**, wherein the source node is an adapter.

33. The source node of claim **26**, wherein the request to transmit data comprises a request to send a frame of data to the target node.

34. The source node of claim **26**, wherein the source node is connected to the target node by a loop and wherein the request to transmit data comprises arbitrating for the loop.

35. The source node of claim **26**, wherein the source node is connected to the target node by a network.

36. The source node of claim **26**, wherein the source node is connected to the target node by a Fibre Channel fabric.

37. The source node of claim **36**, wherein the Fibre Channel fabric is an arbitrated loop.

38. The source node of claim **36**, wherein the Fibre Channel fabric is a switched fabric.

39. A source node for transmitting data to a target node comprising

first determining means, responsive to detecting a request to transmit data to the target node, for activating a timer and detecting an expiration of the timer;

second determining means, responsive to detecting an expiration of the timer, for determining whether space is available in the target node to receive the data; and

generating means, responsive to a determination that space is unavailable in the target node, for generating an indication that the target node is blocked.

40. The source node of claim **39**, wherein the request to transmit data comprises a request to send a frame of data to the target node.

41. The source node of claim **39**, wherein the source node is connected to the target node by a loop and wherein the request to transmit data comprises arbitrating for the loop.

42. The source node of claim **39** further comprising:

halting means, responsive to the indication, for halting attempts to transmit data to the target node.

43. The source node of claim **39**, wherein a determination of determining whether space is available in the target node to receive the data is made by examining buffer to buffer credit from the target node.

44. A source node comprising:

detecting means for detecting a request to transmit data from a source node to a target node;

activating means, responsive to detecting the request to transmit data to the target node, for activating a timer;

monitoring means for monitoring the timer for an expiration of the timer;

determining means, responsive to detecting an expiration of the timer, for determining whether a buffer to buffer credit is present in the target node; and

generating means, responsive to a determination that the buffer to buffer credit is absent, for generating a blocked signal.

45. A computer program product in a computer readable medium in a source node for transmitting data to a target node, wherein the target node has a buffer to buffer credit, the computer program product comprising:

first instructions for detecting a request to transmit data from the source node to a target node;

second instructions, responsive to detecting the request to transmit data to the target node, for activating a timer;

third instructions for monitoring the timer for an expiration of the timer;

fourth instructions, responsive to detecting an expiration of the timer, for determining whether buffer to buffer credit is present in the target node;

fifth instructions, responsive to a determination that buffer to buffer credit is absent, for generating a blocked signal.

* * * * *